REPRINTED FROM

BYTE

October 1981

Local-Area Networks

Possibilities for Personal Computers

Dr Harry J Saal
Nestar Systems Inc
2585 E Bayshore Rd
Palo Alto CA 94303

Today's technical press is filled with announcements of "local-area network" products and "personal computers." New technologies from billion-dollar corporations are being rivaled by products from small firms, in a field no more than a few years old. This article provides an overview of local-area networks and how they relate to personal computers.

Defining local-area network is every bit as difficult as defining personal computer. Features, prices, and technology are distributed across a broad spectrum. Thus, we will try to describe the distinguishing characteristics of a local-area network—how to know one when you see one—and discuss some related system designs that are not local networks, but address many of the same requirements.

Personal Computers and the Group

The revolution in computer systems began with dramatic advances in silicon technology that greatly reduced the cost of the "computing" part of a computer system. Before this, CPU (central processing unit) cycles were a valuable and scarce resource; whole industries grew up developing hardware and software techniques to squeeze out the last bits of efficiency from big mainframes. Learned papers on how to salvage another two percent of processing time dominated computer conferences. People gathered in computer centers (hospital-like environments with air conditioning, raised white floors, and observation galleries). Then, suddenly, all that changed.

The cost of the CPU is no longer the dominant concern. Instead, electromechanical devices such as disks, printers, terminals, and cables generally cost more than the entire central processor. As the prices of these peripheral components drop, the time people spend using the systems becomes more important. We need rapid access to information; we need to review alternatives "online" to make decisions quickly. Our computer systems must respond to our needs and schedules, not the other way around.

The personal computer is dedicated to providing this environment. It is ready for work when we want to use it. It is typically dedicated to one person (or task) and not shared with other people. Although timesharing systems attempted to give the user the illusion of a dedicated computer, they failed because inevitably the load presented by numerous users slowed them down. A personal computer, on the other hand, responds equally well at any time of day. We no longer need to worry about the "wasted cycles" if we simply leave it on our desks just blinking its cursor. The hallmark of the personal computer is this "one person, one computer" approach.

While having to share a central processor may not be justified for many of today's computing needs, information sharing is as important as ever. Once two or more people begin to work cooperatively, they need to communicate and exchange information, whether the impetus be the joint development of a large program, several people checking on information in a common data base, or the implementation of an electronic mail system.

Sharing of larger and more reliable peripheral devices is equally important in all but the smallest computer applications. We can't all have our own letter-quality printer in our office, though we may need access to one. Large libraries of programs or extensive data bases require larger disks than those normally connected to personal computers. Their cost (and reliability) is much higher than

the stripped-down "consumer" variety offered by most personal computers.

Multiuser Systems

Personal computer networks preserve the independence of each computer work station while offering the possibility of sharing information and devices among the individuals on the network. Networks are useful in almost all situations where several people need to work together and share information, but still want the attractive features of the dedicated personal computer.

Of course, we can satisfy the multiuser requirements in a more traditional way, too. A number of companies offer shared multiuser systems based on a single microprocessor. Digital Research's MP/M system permits up to sixteen users to share a common microprocessor and its peripherals. MP/M is a derivative of the popular CP/M operating system that permits applications written for that environment to function for multiple users. Onyx's C8000 is a multiuser system based on the Zilog Z8000 microprocessor running the Western Electric UNIX operating system.

Multiuser systems are fundamentally similar to timesharing systems of the past. Users may be happy with the performance as long as the demands on the single processor are low, but they share one of the great weaknesses of central computer systems in that if the processor should fail, everyone loses his work and has to wait until the system is repaired or restarted. And because of the statistical nature of the sharing of the processor, things we take for granted in personal computers, such as real-time graphics and instantaneous response to keystrokes, are sacrificed.

Networks, Networks, Networks

Until five years ago, a computer-communications network generally meant a connection of a large number of terminals, geographically distributed throughout a company or across the country, to one or more central computers.

Anyone using The Source or Compuserve (Micronet) uses such a network. The terminal is connected by telephone to a nearby communications processor, which takes the fairly low-speed information (30 to 120 characters per second) going to or from your home and merges it with the data of other local users. These communications processors are connected together by much higher speed lines from city to city. The data are put into groups, called *packets*, with routing information and error-detecting fields appended, and sent from site to site until the packets arrive at a processor connected to a large time-sharing system. (Western Union's Telex and TWX services are other examples of international low-speed networks.) These networks process information at speeds appropriate for humans typing or reading data from a screen. Loading a 16 K-byte program or operating system takes over nine minutes at 300 bps (bits per second); the same load would take under one second from a local floppy disk.

Some networks are used to connect computers rather than terminals. They run at much higher speeds and transmit large files, documents, and electronic mail between systems. But even these nets don't have the bandwidth required to allow modern storage devices to operate at full speed, and are not acceptable for the interactive transmission of program and data files in real time.

Device Sharing

There are a number of systems on the market that permit each user to have his own dedicated computer and share disks or printers, but which are not truly computer networks. Several recent products permit numerous independent microprocessor boards and separate memory to be installed in one chassis. One of these boards is generally reserved for shared access to one central disk subsystem or printer. A terminal is attached to each processor, so each user on the system actually has a dedicated microprocessor.

Such a system is quite attractive, but certain inherent disadvantages still remain. The chassis is large, needs a big power supply and large cooling capacity, and if any module fails, the entire system, generally, is down. There is no way to add more stations while the system is running, and the terminals can't be located very far from the main processor unit. Each processor must communicate with the others through the common-service processor. They cannot directly exchange information, nor can they have their own private disks, printers, modems, and the like.

Another product for multiuser, independent-processor sharing of a disk is the *disk multiplexer* (the Corvus Constellation is an example). A disk multiplexer can be likened to a very fast rotary switch that cycles around looking to see if any of the computers connected to it wish to do a disk access. When it finds a request, it reads or writes the particular disk sector and then goes on to the next station. The disk multiplexer approach is quite simple and can be an inexpensive solution for many applications. However, due to the very low level of the requests that are typically presented to the multiplexer (eg: read a sector and write a sector) it is generally limited in dealing with the more sophisticated problems that arise in multiuser interactions.

A more sophisticated interface with a powerful software base is needed for complex applications. Like the multiprocessor systems previously described, there is no way for separate stations to communicate directly. They must send their information to the multiplexer, where it goes to disk, or may be temporarily buffered in memory. If the central disk or multiplexer fails, all work comes to a halt.

Networks Without Software

One of the central themes of a computer network is communications. A large number of companies now offer computer networks that provide the ability to transmit data from station to station, but do not address the questions of the necessary operating system, programming language, and applications software needed to make use of these networks. Basically, these units are peripherals with low-

level drivers that permit data exchange. While they are suitable for those installations that have the necessary system-programming talent to design, modify, and implement the changes needed to take advantage of this facility, we will be focusing on integrated computer-network systems. Very few vendors are willing to step up to the complex software tasks inherent in blending these technologies into a coherent system design.

Both Digital Research and 3COM provide software without a network. Digital Research's CP/NET system permits up to sixteen stations on a host. These stations share the data and devices on that central host. CP/NET is written without any particular network communication devices in mind. Each hardware vendor may select a particular technology and protocol to connect the work stations to the host. But although CP/NET provides a framework for multiuser software based on the familiar CP/M environment, due to the lack of support for applications in the languages and systems running under CP/NET, many companies have chosen to develop their own variant of CP/M with their own sharing protocols.

3COM's UNET is a package written for the UNIX environment. It is a software implementation of a government-standard intercomputer protocol, called TCP; it, too, leaves open the question of how the computers are actually connected, and application programs must explicitly deal with the network in a nontransparent fashion.

Attributes of a Local Network

A local-area network can be described as a communications network that covers a limited geographical area. Just what "limited" means varies substantially, from 0.1 km (approximately 328 feet) to 10 km (approximately 6.2 miles). Data rates on a local network also vary over orders of magnitude, from 100 K bps to 10 M bps, and higher. But these boundaries are far from sufficient to characterize the meaning of "local network" today.

Compared to terminal-like devices, a local network generally has an inexpensive communications medium and high data rates. Every node on the network can communicate with every other node, and the network requires no central node or processor. Messages are "broadcast" over the communications medium, with a destination address included. Only the intended receiver is expected to respond, although other stations have the capability of "listening in." Thus, a high level of security, such as found in point-to-point networks, is not present unless cryptographic techniques are used. Local networks are meant to be highly reliable, so that any failing station will simply be unavailable, without interrupting the communications between the remaining stations. Similarly, it is possible to add new stations without disrupting the ongoing communications flow.

Due to the limited-distance nature of local networks, another standard feature is the ability to connect multiple networks. This internetwork link, called a *gateway*, may be a high-speed link for networks that are close to each other, or it may depend on a more conventional telecommunications network for reliably transmitting data from city to city, or around the world. Because of the multiplicity of emerging network technology, and the variety of communications protocols in use, gateways must be provided to permit stations on one type of network to exchange information with others on a different type or speed of network. Both electrical and software protocols must be converted when passing data through these gateways.

Origins of Local-Area Networks

Local-area networks evolved from the large-scale telecommunications networks developed in the 1960s. As universities and research labs began to install computers, the need arose to permit the flow of information among them. The underlying communications protocols (packet transmission) came from the long-distance networks. The communications media (twisted pair or coaxial cable) were developed to support very high speed direct coupling between computers.

One experiment significantly affected the nature of modern local-area networks: the University of Hawaii wanted to connect terminals all over the Hawaiian islands to a local computer and communications processor, and from there to other networks. They developed a system called ALOHA, a packet radio-transmission system. No wires were used to connect each station to the others, so techniques such as polling could not be used.

The scheme was elegant, and operated in a manner very similar to the way that telephone party lines work. Each station would first listen to see if anyone else was transmitting (in radio jargon, this was called "carrier sense"). If not, the station would transmit its message, including error-detection bits. As long as the total fraction of available transmission time used was low, everyone got a turn—eventually. If two stations found the channel clear and started transmitting simultaneously, the two packets would *collide*. This collision would scramble the information, but the error-detection logic would throw away the bad data. If the stations didn't receive an acknowledgment by a certain time, they would simply send the packet again.

Studies of this scheme quickly revealed a number of problems, one of the more serious being that as the number of messages grew, many collided, and only a small fraction of the true communications bandwidth was used for valid data. Far more serious was the fact that if enough stations tried to transmit, less and less data got through, and the result was continuous collisions!

The Ethernet Network

Numerous refinements to the basic

ALOHA scheme were developed, but the most significant were developed at the Xerox Palo Alto Research Center as part of an experimental project, called Ethernet, started in the mid 1970s. (It was once thought that a universal medium called "luminiferous ether" was the carrier of electromagnetic waves. Xerox decided to build its "ether" out of coaxial cable.)

The Ethernet scheme could detect a collision in progress by reading back the state of the cable *as data were being transmitted*. Thus, a station could sense when another station was sending data and stop transmitting, instead of continuing until the end of its packet. (To guarantee that all such stations recognized the collision, a burst of noise was sent prior to quitting.) A randomized delay function was added so that each station would wait a different amount of time, instead of beginning to transmit immediately after a previous transmission was complete. This avoided causing a collision each time two or more stations had something to send. The delays would get progressively longer as the channel became busier.

Using these modifications, an Ethernet-style local network could use essentially all the bandwidth of the communication medium. Even as stations began sending ten times as much information as the channel could handle, things no longer came to a halt.

The Ethernet algorithms were designed to be simple. Every station on the network manages its use independently, so there is no need for a master to control access. Simplicity was important to ensure minimal building costs and reliability. Other schemes are considerably more complex, which makes them either difficult or expensive to include in each node's interface.

Network Topology

Most early local networks used a *star* topology (see figure 1): a central node was connected via a radial cable to each of the other stations. Unfortunately, this system suffers from the consequences of a central failure. The entire system goes down if the center fails. But there are still many reasons to use a star network. Telephone exchanges are organized as star networks, and many companies already have PBXs (private-branch exchanges). By using the PBX as a local-area network for data as well as voice communication, companies can take advantage of the already existing wiring: this is most suitable for low-data-rate information, such as video terminals.

A *ring* (or *loop*) topology connects its stations in a closed network. Messages circulate in one direction, often being amplified and repeated at each node they pass through. Again, a station failure can interrupt the entire message flow, but in some cases two alternate parallel loops are provided for reasons of reliability. Rings often use a form of control strategy called a *token*. A token is a special message that gives the receiving station permission to transmit. When a free token comes by a station that wants to transmit, the token is removed and replaced by the message. Generally, the same station removes this message when it comes around again and reinserts the token.

Rings are most popular in process-control applications (eg: controlling equipment in manufacturing environments). When dealing with the equipment being controlled, it is important to be able to guarantee the worst-case maximum time necessary to send a message to some station, say to close a valve. Token systems can provide a solution to this problem. The random nature of the Ethernet scheme might prevent a station from sending a critical message in time. (Actually this is a bit misleading. Ethernet can be used to build token-like control that requires stations to avoid sending a message just because they see the net is free; they have to wait to receive the control token first.)

Much of the ring approach has been developed in England, particularly at Cambridge University, where numerous computers and terminals have been interconnected using a simple but high-speed interface. Several British companies are now developing commercial versions of the Cambridge Ring interface unit.

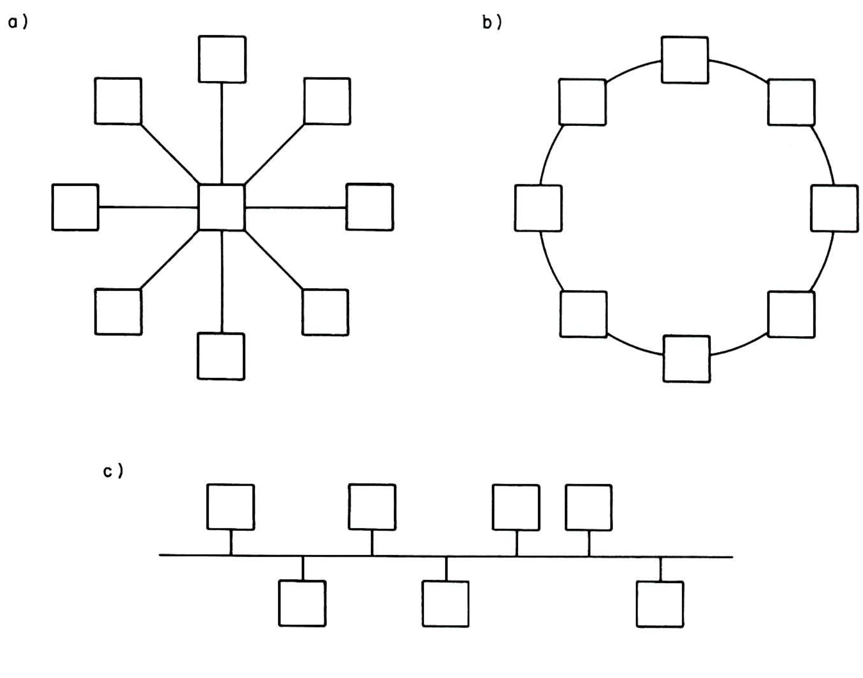

Figure 1: *Popular network topologies. The star network (1a) is the most common of the early network types (such as the telephone system), and relies on the central node for control of operations. The ring network (1b) circulates all messages in one direction, and may employ tokens to specify which node may transmit; a failure of any node may interrupt network operation. Bus configuration (1c), as used on the Ethernet and by cable television, allows nodes to be added or removed without impairing the network.*

Bus topology is quite simple, being merely a long length of cable that runs past each station. Stations are connected to it at the nearest point, and can be added or removed without affecting any other station. A station can be added in two ways: the bus can be split, temporarily disrupting communications, and a new station inserted, or, more commonly, *taps* (devices developed by the cable TV industry that literally pierce the cable from the outside, making contact to the inner conductor and the outer shield) can be installed while stations are transmitting. Even temporary shorts will only garble some packets, and they will be retransmitted once the short is removed. The Ethernet uses this form of interconnection.

What Frequency, What Wire?

Another significant parameter in the description of a local-area network is the particular medium used to send the information between stations. Local networks have used twisted pair, multiconductor flat cable, coaxial cable, optical fibers, and even infrared light transmitted through the atmosphere. Within each of these categories, numerous choices abound in the frequency used for transmission and the details of the modulation technique.

The most fundamental split in technology revolves around frequencies used on coaxial cable. You can think of coaxial cable as a simple wire. If someone wants to send information, the wire can be left at 0 V or raised to some nonzero voltage. Another station can detect the changes in voltage and decode the information. This is generally referred to as *baseband transmission*, since the frequency spectrum generated starts at 0 Hz (direct current) and goes up from there.

Television transmission is sent at very high frequencies (typically 50 MHz to 100 MHz). A central carrier-frequency is modulated up and down to transmit the information. At these frequencies, the cable has far less attenuation than in the baseband region, so a transmitter can broadcast over miles of cable instead of being limited to several thousand feet. And the blossoming cable-TV industry can provide the necessary devices at a very low cost due to the large volumes they are expected to produce for standard television reception. RF (radio-frequency)-modulated systems can also provide much higher bandwidths than baseband, so the cable can, in principle, be shared along with voice- and video-transmission systems.

RF systems (also known as *broadband*), while very attractive, do require a central retransmitter to receive the data sent from each station and rebroadcast it, much amplified, at a different frequency that each station is expected to listen on. The required unit is expensive, even for the smallest system, and if that unit fails, the network is unavailable until the retransmitter is back in service.

Local-Area Network Standards

Numerous local-area network products have already been announced, and new entries are made daily. In almost every case, the manufacturers have developed their own hardware and software protocols. These, naturally, are incompatible with everyone else's!

The exception to the above incompatibility is the Ethernet specification released in November 1980 by DEC (Digital Equipment Corporation), Intel, and Xerox Corporation. Based on years of actual experience with an experimental version of Ethernet communications, the "tri-company standard" was provided, with every detail of the electrical and low-level communications protocols defined. These companies are trying to encourage the adoption of this scheme among computer and peripheral manufacturers; indeed, many large and small companies have publicly announced their adoption of the DIX Ethernet system, and are busy designing and building products.

The DIX Ethernet system uses a baseband-transmission scheme, with a 10 Mbps data rate. It provides for the use of a large number of stations and packet formats, with 48 bits allocated for a unique world-wide station address that is not duplicated anywhere, and it has a large (32-bit) checksum on each packet to detect errors.

This scheme pushes the technological requirements by operating at such high speeds and using the particular packet format and checksums adopted. Without specially designed VLSI (very large scale integration) devices to handle the network interface, it is expensive to build an Ethernet interface. For example, Intel has announced a Multibus Ethernet interface (the iSBC-550) that costs about $4000. To that you must add several hundred dollars for an analog interface (the transceiver unit) to connect between the interface board and the physical cable. It is expected that volume production of the needed components will begin within the next two years and prices will drop dramatically.

One means of lowering the effective cost is to share the Ethernet interface among several stations. A number of companies (such as Xerox, and Ungermann-Bass) offer a microcomputer-based Ethernet interface with four to eight ports for connecting terminals or other microcomputers. The effective cost per station can be reduced to between $500 and $1000 for a fully loaded system.

Standards Organizations

While product activity continues, several committees are attempting to develop an industry-wide standard for local-area networks. The IEEE (Institute of Electrical and Electronics Engineers) Computer Society Local-Network Committee (Project 802) has been meeting for over a year to try to establish a viable standard, and the standard is still in a state of flux. Fierce battles have been raging among the committee members representing different local-network interests. The IEEE standard has been evolving in a manner that attempts to accommodate many diverse application areas and functional requirements.

The framework for defining a communications network is based on a

highly layered series of protocols developed by the ISO (International Standards Organization), called the OSI (Open System Interconnection) protocols. The OSI architecture defines seven layers of communications.

Layer 7, the Application layer, provides for the identification of users and services, and is responsible for initiation and reliability of data transfers, as well as general network access, flow control and recovery. Utility programs may perform network file-transfers, terminal-to-network support, etc.

Layer 6, the Presentation layer, is primarily responsible for making data available to the Application layer in a meaningful fashion. The Presentation layer takes care of protocol conversion, data unpacking, translation, or encryption.

Layer 5, the Session layer, is used to set up and break communications paths across the network and manage the exchange of data. It is responsible for multiplexing and demultiplexing messages, managing the sequencing and priority of these messages, and providing the needed buffers.

Layer 4, the Transport layer, provides another level of connections between network entities. This layer manages the connections and segments messages into smaller pieces that the network can support. It may also be involved in error and flow control, as well as additional multiplexing activities.

Layer 3, the Network layer, is the level that actually determines how to get a message from one network to another (since many paths may exist). The Network level may use several intermediate hops to get information to its ultimate destination and, thus, needs to know how to route packets through the network. It, too, may be involved in sequencing and error- and flow-control activities.

Layer 2, the Data-Link layer, is where the actual packet formats are established, along with the particular access control mechanism used to regulate use of the physical network. Data is encapsulated in packets that contain physical addressing information, error-detecting checksums, etc.

Layer 1, the Physical layer, defines the electrical and mechanical interfaces to the network. The Physical layer specifies the particular signaling means (baseband vs RF, for instance), the modulation technique adopted, station-identification addresses, etc.

The current activity of the IEEE 802 committee is focused on specifying Layers 1 and 2, the Link and Physical levels. Similarly, the DEC/Intel/Xerox Ethernet specification addresses only these two levels of protocol.

It appears that the 802 Committee is converging on a standard that offers many alternatives within one framework. Even the issue of data rate (specified by Ethernet as 10 M bps) appears to be an optional value (such as 1, 5, 10, or 20 M bps). The error detection used may be either a 16- or 32-bit CRC (cyclic redundancy check) code, and the access method may be either a token-like scheme or a CSMA/CD (carrier-sense, multiple access with collision detect) scheme resembling (but not identical to) the Ethernet system. While the 802 Committee deliberates, manufacturers continue to develop their own systems. It is possible that some may modify their products once standards activities are resolved.

Recently, attention has been given to the higher levels of protocols. The National Bureau of Standards is proposing a series of Transport and higher-level protocols. It is unfortunate that the work on the higher-level protocols does not precede the lowest-level issues. The advantage of layered protocols is that the underlying levels can be changed in ways transparent to the higher levels, while the converse is not true, but the standards activities are not moving in that direction.

Servers and Clients

The most significant contribution in the local-area network field is not the communications aspect, but the

development of a whole new way of building computer systems. The fundamental organization described by Xerox assumes a fully distributed control mechanism (see figure 2). There is no master-slave relationship among stations; they all communicate and cooperate with one another. Any number of stations (called *servers*) on a local network may provide services to other stations (called *clients*). Typical server functions are: mass-storage file system, printer support, time-of-day clock, translation of symbolic names into physical addresses, data-base management support, gateways to other networks or computers, and other specialized hardware support. Servers may also be clients of other servers on the network. For instance, the printer server may be a client of the file-system server in the course of serving its own clients.

Servers are distinguished on the network merely by the software they run and any special hardware they contain. A station that is willing to listen to requests from other stations (using a higher-level protocol they agree on) can perform a server function.

In order to maintain a high level of reliability, the logical functions of the servers are usually implemented using separate physical computers. One could merge all of the above services into one larger computer, but in doing so would end up with something resembling a conventional central computer system.

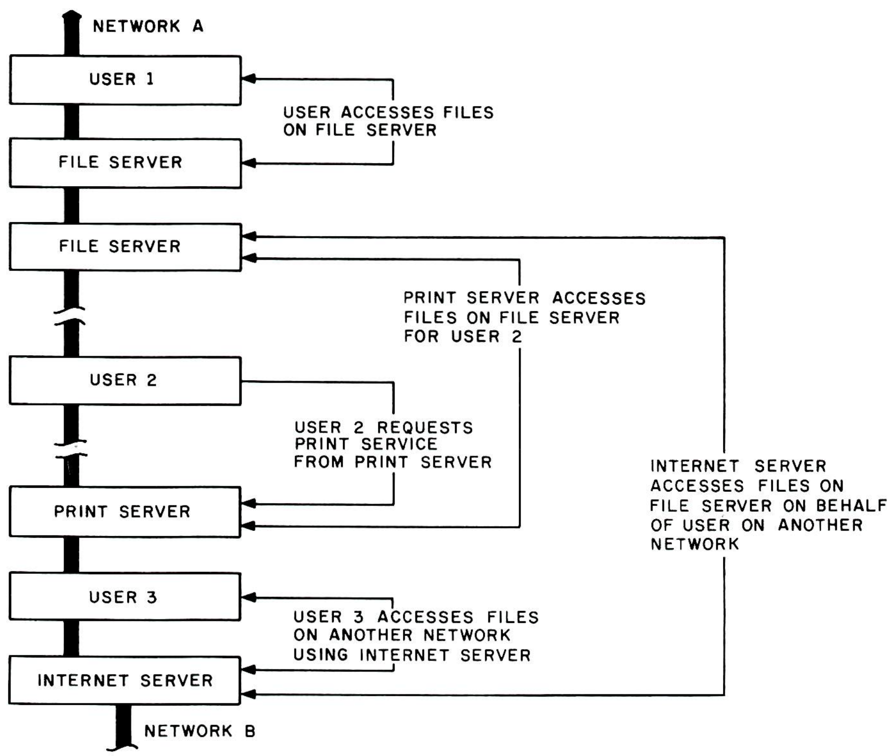

Figure 2: *The server/client relationship on local networks. Perhaps the most significant advance in communications is that, under this scheme, the computer system is fully distributed; there is no master node, so each node can call on others when resources are needed. Some nodes are dedicated to special functions, such as controlling hard-disk mass-storage devices, or printing.*

Putting It All Together

Clearly, the local-area network field is too broad to cover in great depth. Most of the attention has focused on nonpersonal computer systems, such as large mainframes or terminals. We will describe the Nestar Cluster/One Model A system.

The Cluster/One Model A is a local-area network system based on the Ethernet principles, but its implementation has been optimized for the connection of low-cost Apple II personal computers. The system was first announced in January 1980 and has been used around the world for almost two years. It includes integrated software and hardware features needed to provide a comprehensive data-processing and data-communications facility, and the system permits either independent operation of individual stations, with a full complement of local peripherals, or a share in the larger, more reliable peripherals via the local network. The work station in question costs between $1000 and $2000, so cost constraints differ from those applied to networking work stations in the $10,000 to $20,000 price range. Nestar chose to implement many network functions via simple programmable hardware, and assigned many functions to software. Another decision influenced by these cost factors involves network speed. The speed of the Cluster/One was decided by the reasonable cost for a personal computer network interface and the bandwidth requirements needed for the work typically done by these personal computer work stations.

The Model A network operates at 240 kbps—almost a thousand times faster than a 300-bps telephone link, and 40 times slower than the Xerox Ethernet system. This was the fastest rate that could be supported by network-interface software running on the 6502 host processor of the Apple II computer and still allow data checksums to be performed on the message packets.

The choice of the network medium was also influenced by the basic cost goal. Rather than taking 8-bit data from the Apple memory and then serializing and deserializing it, it was decided to transmit the data in an 8-bit-wide parallel fashion, which not only reduced the interface cost, but increased the inter-bit transition time. This has the effect of permitting essentially arbitrary interconnection topology for the Nestar network, something not found in any other system. The Nestar network is not restricted to a linear-bus topology, but can be wired as suits the particular installation requirements.

Network Design

The overall system design resembles the Ethernet scheme. No single critical component must func-

tion for network communication to take place. All station-to-station communication is direct, with a carrier-sense algorithm executing in the ROM- (read-only memory) based protocols in each station interface. The interface is passive, so stations may be added or removed from the network during operation. Stations not in use may be turned off until needed.

In the Model A network, the carrier-detect function is implemented using a dedicated control line, which indicates the bus is busy. Stations do not transmit until they see that this line is available. The electronics of the bus interface permit reading of data just written. However, it is not necessary to perform full collision detection. At the start of a packet transmission the address of the station attempting to send is first put on the bus, and then read back. If two stations do this simultaneously, at least one will not read back its own address and will detect a conflict. Even this is rare, since each station has a random waiting algorithm that avoids most collisions that would occur at the end of a previous transmission. Once this initial collision detection has been passed, the carrier signal has been established and further collision detection is not necessary. The rest of the packet is sent, like ALOHA, without collision detection. After the initial check, later collisions can result only from erroneous stations, and not under normal conditions.

Each packet of data contains initial header information, followed by up to 256 bytes of data and a 16-bit checksum. Once the packet is transmitted, the receiving station immediately acknowledges the receipt of the packet (if the checksum matches the data) or else requests a retransmission. This error-control algorithm is completely contained in the ROM-based protocols on the Nestar interface, and permits higher levels of software to work with reliable and correctly sequenced data. The ROM protocols are also responsible for taking messages longer than the 256-byte packet size and splitting them into multiple packets, each with its own checksum. Thus the four lowest layers of the OSI protocols are supplied as part of the logic on board the Nestar network interface.

The Model A network also includes a variety of network servers and the software needed to make their use literally transparent to current applications. The Nestar Network File Server runs on an Apple II microcomputer interfaced to the network. It can support a variety of devices, ranging from two 8-inch double-sided floppy disks, to 66 megabytes of hard-disk storage. Larger capacity is available by using more than one file server. The network software allows multiple file servers on one local network, thus giving essentially unlimited online storage capacity. The data on these reliable, sealed Winchester disks can be "backed up" using Nestar's compact cartridge-tape streamer drive. A single cartridge can write and check over 20 megabytes of data in twelve minutes.

The Network File Server can also contain a real-time clock/calendar, which stations can interrogate. This facility is used to *timestamp* the creation, access, modification, and backup times of network files. Files are organized with a tree-structured system similar to a UNIX directory; they can be password-protected in a variety of ways to ensure that only authorized users can create, modify, or otherwise access network data.

The software provided makes the use of this data straightforward from any Apple II work station on the network. All of Apple's current operating systems (DOS 3.2, DOS 3.3, Pascal 1.0, and Pascal 1.1) can be directly loaded over the network. Modifications are made during this process so that stations can logically connect to virtual disks on the network shared disks (either from keyboard commands or from programs). These disks need not have the same capacity as 5-inch floppy disks, but may be much larger or smaller. Each storage area is allocated the appropriate size for the application; users may be executing programs in any set of languages or operating systems at the same time.

Network Applications

The Cluster/One network has been used in a variety of applications that include general office-automation environments, engineering and software development sites, educational and entertainment uses, and special turnkey applications, such as travel-agency and real-estate systems.

To support this variety of uses, Nestar provides a number of general-purpose computing products. Other servers, such as print servers supporting a multiplicity of printers, are available. Communications servers support internetwork activity. Application programs for general database access, interoffice electronic mail, and teleconferencing, have been developed by Nestar, either in-house or in conjunction with the suppliers of popular packages for the Apple II. The collection of hardware and software capabilities makes this network attractive for a wide range of application areas.

What's Next?

There seems to be little doubt that the current interest in local-area networks and personal computer work stations will continue to grow over the next few years. As stations become more powerful and sophisticated in both systems software and applications programming, they will replace an even larger fraction of conventional minicomputer systems. As manufacturers provide fully integrated VLSI components designed for very high performance networks, they will be incorporated into the personal computer local-area network interfaces. Whether or not the standards activities will stem the proliferation of de facto standards remains to be seen. The emergence of networks of personal computers has opened up a whole new set of challenges for programmers in developing real-time, multiuser, interactive systems. ■